

# ALBA MUSICA

## 1. Abstract

*Alba Musica* è una casa discografica indipendente italiana, con sede a Roma e molto attenta sia alla valorizzazione di nuovi artisti sia alla costante affermazione sul mercato di gruppi o solisti già avviati. Al contrario di molte altre case discografiche, *Alba Musica* presenta tra i suoi assistiti artisti appartenenti a diverse categorie musicali, dal Rap al Pop, dall'Indie al Rock, passando anche per la Techno-House. Proprio per questa sua generalizzazione, sono molti gli artisti novizi che scelgono *Alba Musica* come loro prima etichetta, perché questo consente loro di poter sperimentare la propria musica per trovare la giusta quadratura e specializzarsi poi su un determinato genere, magari con un successivo contratto con una major più esigente e restrittiva. L'aumento esponenziale di assistiti ha reso quindi necessaria la realizzazione di una base di dati chiara e organizzata per la gestione dei clienti, con tutti i brani e gli album registrati presso gli studi dell'etichetta, categorizzati in base al genere, e i collegamenti con produttori esterni per la commercializzazione, sia fisica sia telematica. La gestione dei dipendenti viene invece organizzata in un database a parte, creato al momento della nascita dell'etichetta discografica.

## 2. Analisi dei requisiti

### 2.1 Descrizione dei requisiti

Si vuole realizzare una base di dati per la gestione di una casa discografica. La casa discografica si occupa della registrazione delle tracce e della seguente distribuzione commerciale.

L'entità principale da considerare è la **Traccia**, una registrazione identificata da una denominazione. Ha inoltre una durata e una data di registrazione.

Una o un insieme di tracce compone un **Brano**, identificato un titolo. Ha una durata, una data di composizione ed una lista di tag (esempi di tag sono: explicit, aggressivo, relax, chill etc.).

Un Brano, "singolo", oppure un insieme di canzoni, "album", hanno valore commerciale e vengono a tutti gli effetti considerati prodotti. L'entità **Prodotto** è identificata da un titolo che sarà uguale al titolo del brano nel caso esso sia un singolo oppure sarà il titolo dell'album. Ha inoltre una durata, una data di pubblicazione ed una copertina. Nel caso dell'album abbiamo anche il numero di brani.

Il Brano ed il Prodotto appartengono ad un **Genere musicale**, esso è identificato da un nome e può essere considerato sottogenere di un altro genere musicale.

Una Traccia è il risultato del lavoro di tre principali tipologie di artisti:

- **Produttore**: produce la base musicale per la traccia
- **Scrittore**: scrive il testo, qualora presente, cantato nella canzone
- **Cantante**: canta il testo della canzone

Un singolo pezzo può essere prodotto, scritto e cantato da una singola persona oppure da più artisti di tipologie diverse.

Un **Artista** è identificato dal suo nome d'arte e può essere solista oppure un gruppo musicale. Il gruppo musicale è identificato da un nome d'arte ed è composto da 1 o più solisti. Il solista è identificato da un nome d'arte ed ha gli attributi: nome, cognome e data di nascita.

Un Artista per poter usufruire degli studi musicali dovrà effettuare una Prenotazione, a seconda del suo contratto dovrà poi effettuare o meno un pagamento. Una prenotazione può quindi determinare una ricevuta. **Prenotazione** è identificata dall'Artista, lo studio e la data di inizio della sessione; ha inoltre una durata.

Una prenotazione quindi occuperà uno Studio nel quale viene registrata una Traccia. **Studio** è identificato da un nome, ha inoltre un costo orario ed un indirizzo fisico.

Dalla traccia, dal brano e dal prodotto derivano dei **Diritti d'autore**, identificati dal codice univoco DOI. Il diritto d'autore può essere considerato l'insieme di due componenti diverse: il **Diritto morale**, derivato dalla canzone vista come proprietà intellettuale, ed il **Diritto commerciale**, derivato dal prodotto commercializzabile. Il Diritto d'autore è detenuto totalmente dagli artisti. Il Diritto morale deriva dall'esistenza della Traccia e del Brano mentre il Diritto commerciale esiste quando esiste un Prodotto commerciale. Quest'ultimo viene concesso temporaneamente ad un'azienda che gestisce la

distribuzione, Distributore. La concessione non ha una durata prestabilita e il ricavo ottenuto è stimato in base alla data di inizio della concessione ed al suo costo mensile.

Alba musica intraprende una relazione commerciale con diverse Aziende, l'entità **Azienda** è identificata da un codice univoco di fatturazione e ha un nome, email e numero di telefono.

Un'Azienda può essere un **Produttore** oppure un **Distributore**.

Un Distributore, dopo aver noleggiato i diritti commerciali, si occupa della distribuzione telematica e/o fisica del Prodotto. La distribuzione telematica è gestita dall'**Ente telematico** ed i ricavi degli artisti derivati dagli ascolti vengono versati agli autori del prodotto dall'ente telematico. Nel caso dell'**Editore** Alba musica ottiene un ricavo dalla quantità di copie fisiche vendute ad esso, la seguente distribuzione nei negozi è a carico dell'editore.

Un produttore si occupa della fabbricazione di un **Prodotto Fisico** a partire dal Prodotto commercializzabile. Il Prodotto fisico ha un tipo (es. cd, vinile etc.), un costo di produzione, un costo di vendita e la percentuale di ricavo destinata agli artisti. La quantità di copie prodotto è definita nella relazione tra Produttore e Prodotto fisico. Il prodotto fisico viene quindi venduto ad uno o più Distributori.

La nostra realtà ha per cui un rapporto finanziario con le Aziende e gli Artisti. Essi sono entrambi clienti che ricevono una retribuzione o che effettuano un pagamento. L'entità **Cliente** è identificata da un codice univoco id.

Un cliente stipula un **Contratto** identificato da un id. Ha inoltre un tipo di contratto (es. Concessione diritti d'autore, usufrutto studi di registrazione etc.), una data di inizio, una data di termine e una descrizione.

Il contratto definisce quindi una **Ricevuta**, identificata da un id. Ha una descrizione, una data di emissione ed un importo.

## 2.2 Glossario dei termini

Termine	Descrizione	Collegamento
Traccia	Registrazione musicale.	Studio, Brano, Artista, Diritto morale
Brano	Insieme di tracce musicali che formano un brano musicale.	Traccia, Genere musicale, Prodotto, Diritto morale
Tag	Tag musicali (es. chill, relax, explicit etc.).	Attributo di Brano
Genere musicale	Genere musicale.	Brano, Prodotto
Prodotto	Brano (singolo) o insieme di brani (album) con valore commerciale.	Padre di Album e Singolo. Diritto economico, Distributore, Prodotto fisico, Genere musicale
Produttore	Azienda che si occupa della fabbricazione di un prodotto fisico.	Figlio di Azienda. Prodotto fisico
Prodotto fisico	Album o singolo commercializzabili come mezzo fisico (es. cd, vinile etc.).	Editore, Produttore, Album
Distributore	Azienda che si occupa della commercializzazione di un prodotto fisico o della distribuzione telematica.	Padre di Editore ed Ente telematico. Diritto economico, Prodotto
Azienda	Azienda che intraprende una relazione commerciale.	Figlio di Cliente, Padre di Produttore e Distributore.
Diritto d'autore	Diritto d'autore su tracce, brani e prodotti commerciali.	Padre di Diritto economico e Diritto morale. Artista
DOI	Digital Object Identifier, identifica il diritto d'autore registrato.	Attributo di Diritto economico
Diritto economico	Diritto d'autore legato alla commercializzazione.	Figlio di Diritto d'autore. Distributore, Prodotto
Diritto morale	Diritto d'autore inteso come proprietà intellettuale.	Figlio di Diritto d'autore. Traccia, Brano
Artista	Artista che può produrre, scrivere e/o cantare una traccia musicale.	Figlio di Cliente, Padre di Gruppo e Solista. Traccia, Prenotazione, Diritto d'autore.

Prodotta	Artista che produce una base musicale per la Traccia.	Relazione tra Artista e Traccia
Scritta	Artista che scrive il testo per una Traccia.	Relazione tra Artista e Traccia
Cantata	Artista che produce una componente vocale per una Traccia.	Relazione tra Artista e Traccia
Cliente	Entità che intraprende una relazione commerciale.	Padre di Azienda e Artista. Contratto
Ricevuta	Ricevuta in entrata o uscita.	Contratto, Prenotazione
Contratto	Contratto stipulato da un cliente.	Ricevuta, Cliente
Studio	Studio di registrazione.	Traccia, Prenotazione
Prenotazione	Prenotazione di una sessione in uno studio di registrazione.	Ricevuta, Artista, Studio

## 3. Progettazione concettuale

### 3.1 Lista delle Entità

Elenca le classi che formano la base di dati. Gli identificatori sono sottolineati.

- **Traccia:** rappresenta una singola registrazione musicale
  - denominazione: varchar (100) primary key
  - durata: time
  - data\_registrazione: date
- **Brano:** indica una canzone completa, formata da una o più "tracce"
  - titolo: varchar (50) primary key
  - durata: time
  - data\_composizione: date
  - tag: varchar (15)
- **Prodotto:** rappresenta il valore commerciale della canzone.
  - titolo: varchar (50) primary key
  - data\_publicazione: date
  - copertina: varchar (100)
  - durata: time

L'entità Prodotto si specializza in due tipologie di prodotti tramite una generalizzazione totale:

- **Album:** è l'insieme di più canzoni o più prodotti "singoli".
  - numero\_branzi: int
- **Singolo:** è la commercializzazione di una sola canzone.
- **Genere musicale:** indica il genere musicale di una traccia; un genere musicale può essere padre di un altro genere (Sottogenere).
  - nome: varchar (30) primary key
- **Diritto d'autore:** rappresenta il diritto d'autore derivato da un pezzo musicale.
  - id: varchar (100) primary key

L'entità Diritto d'autore viene suddivisa in due tipi tramite generalizzazione parziale:

- **Diritto economico:** rappresenta il diritto all'uso commerciale ed è cedibile a terzi.
  - DQI: varchar (100) primary key
- **Diritto morale:** rappresenta la proprietà intellettuale appartenente esclusivamente all'artista.

- **Cliente:** è un attore che può avere una relazione commerciale con l'azienda.

◦ id: int primary key

L'entità Cliente si specializza in due tipi tramite generalizzazione totale:

- **Azienda:** rappresenta un'azienda di terze parti che ha rapporti con la nostra etichetta per quanto concerne la produzione o la distribuzione commerciale delle opere.
  - codice\_univoco\_fatturazione: varchar (100) primary key
  - nome: varchar (100)
  - telefono: char (20)
  - e-mail: varchar (30)

L'entità Azienda è una generalizzazione totale di:

- **Produttore:** l'ente che si occupa della produzione fisica dei dischi.
- **Distributore:** un'azienda che usa un mezzo fisico o telematico per la distribuzione.

A sua volta, l'entità Distributore è una generalizzazione parziale di:

- **Editore:** l'azienda che si occupa della commercializzazione delle sole copie fisiche.
- **Ente telematico:** Quei canali che si occupano della commercializzazione radiofonica o via streaming.
- **Artista:** rappresenta il produttore e/o scrittore e/o cantante di una canzone.

- nome\_d'arte: varchar (30)
- L'entità Artista è una generalizzazione totale di:
  - **Solista**: rappresenta l'artista come soggetto singolo.
    - data\_nascita: date
    - nome: varchar (20)
    - cognome: varchar (20)
  - **Gruppo musicale**: rappresenta l'artista come insieme di soggetti o di "solisti".
    - n\_membri: int
- **Prodotto fisico**: indica le copie fisiche dei dischi, realizzate dal produttore e pronte per la vendita.
  - tipo: varchar (20)
  - costo\_produzione\_per\_copia: money
  - costo\_vendita: money
  - percentuale\_ricavo: decimal (0,2)
- **Studio**: rappresenta uno studio di registrazione in cui vengono sviluppate le varie tracce.
  - nome: varchar (40) primary key
  - costo\_orario: money
  - indirizzo
    - via: varchar (100)
    - civico: varchar (8)
    - CAP: char (5)
    - stato: char (2)
- **Prenotazione**: indica una serie di dati utili ad organizzare la gestione degli studi di registrazione.
  - studio: int primary key
  - nome\_artista: varchar (30) primary key
  - data: date primary key
  - durata: time
- **Contratto**: rappresenta le diverse promesse di prestazioni tra la casa discografica e i vari clienti.
  - id: int primary key
  - tipo: varchar (30)
  - descrizione: varchar (200)
  - data\_inizio: date
  - data\_termine: date
- **Ricevuta**: rappresenta le fatture emesse e ricevute in seguito alla risoluzione dei vari contratti.
  - id: int primary key
  - descrizione: varchar (200)
  - data\_emissione: timestamp
  - importo: money

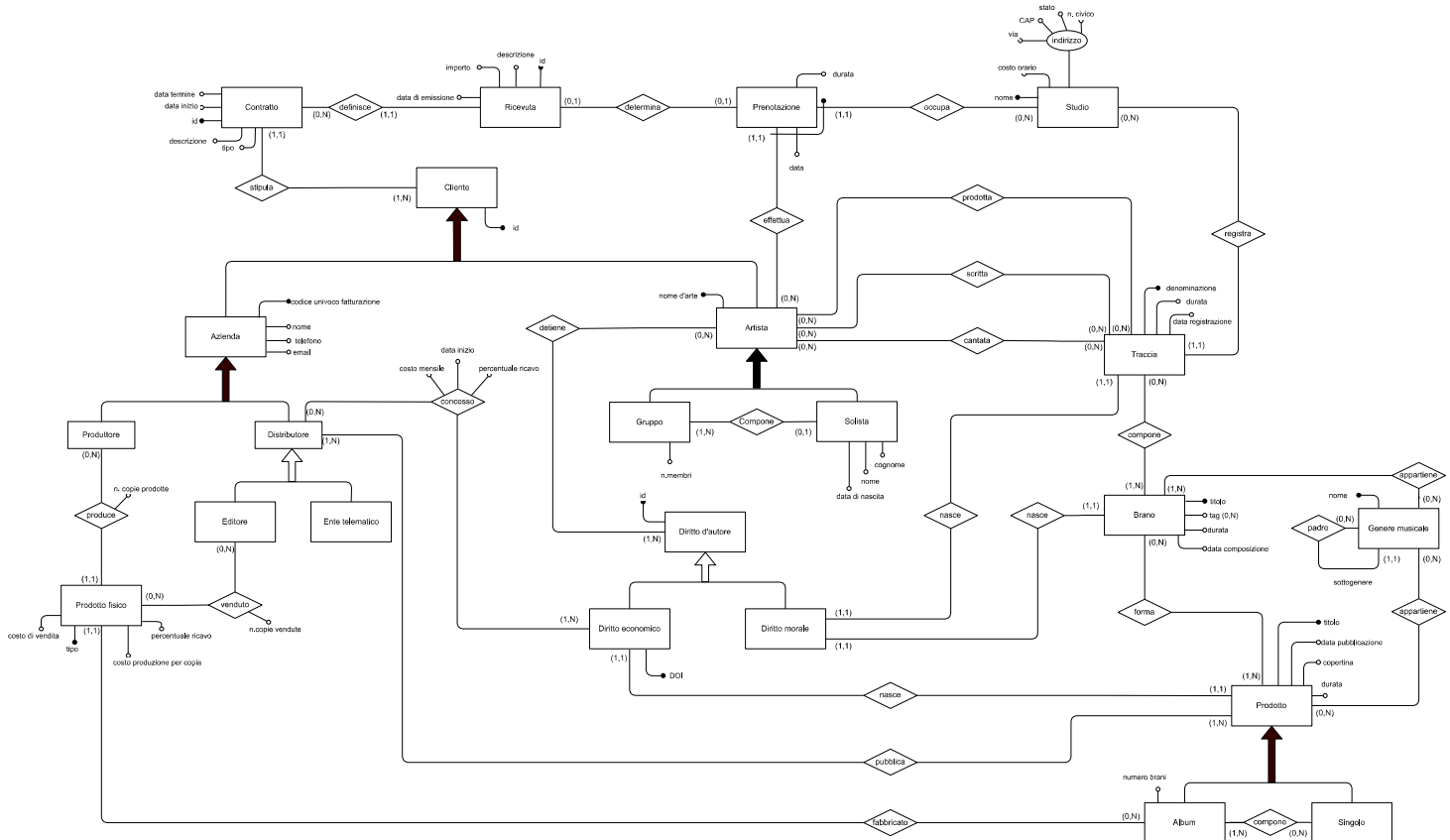
### 3.2 Lista delle associazioni

Relazione	Entità coinvolte	Descrizione	Attributi
compone	Traccia(0,N) Brano(1,N)	Una o più tracce compongono un brano. Un brano è composto da una o più tracce.	Nessun attributo
forma	Brano(0,N) Prodotto(1,N)	Un brano forma zero o più prodotti Un prodotto è formato uno o più brani	Nessun attributo
compone	Album(1,N) Singolo(0,N)	Un album è composto da 1 o più singoli Un singolo può comporre zero o più album	Nessun attributo
appartiene	Genere musicale(0,N) Prodotto(0,N)	Un prodotto può appartenere a zero o più generi musicali Un genere musicale può rappresentare zero o più prodotti	Nessun attributo
appartiene	Brano(1,N) Genere musicale(0,N)	Un brano può appartenere a uno o più generi musicali Un genere musicale può rappresentare zero o più brani	Nessun attributo
padre	Genere musicale(0,N)	Un genere musicale è padre di zero o più sottogeneri, mentre un genere musicale è	Nessun attributo

	Genere musicale(1,1)	sottogenere di 1 solo genere musicale padre	
prodotta	Traccia(0,N) Artista(0,N)	Una traccia può essere prodotta da zero o più artisti e un artista può produrre zero o più tracce	Nessun attributo
scritta	Traccia(0,N) Artista(0,N)	Una traccia può essere scritta da zero o più artisti e un artista può produrre zero o più tracce	Nessun attributo
cantata	Traccia(0,N) Artista(0,N)	Una traccia può essere cantata da zero o più artisti e un artista può produrre zero o più tracce	Nessun attributo
nasce	Brano(1,1) Diritto morale(1,1)	La stesura di un brano implica la nascita di un solo diritto morale Uno specifico diritto morale è derivabile da un solo brano	Nessun attributo
nasce	Traccia(1,1) Diritto morale(1,1)	La registrazione di una traccia implica la nascita di un solo diritto morale Uno specifico diritto morale è derivabile da una sola traccia	Nessun attributo
nasce	Prodotto(1,1) Diritto economico(1,1)	La realizzazione di un prodotto implica la nascita di un diritto economico Uno specifico diritto economico è associabile ad un solo prodotto	Nessun attributo
detiene	Artista(0,N) Diritto d'autore(1,N)	Un artista detiene zero o più diritti d'autore Uno specifico diritto d'autore può essere detenuto da uno o più artisti	Nessun attributo
pubblica	Prodotto(1,N) Distributore(1,N)	Un distributore pubblica uno o più prodotti Un prodotto è distribuito da uno o più distributori	Nessun attributo
fabbricato	Prodotto fisico(1,1) Album(0,N)	Un produttore fabbrica uno o più prodotti Un album può essere fabbricato da zero o più produttori	Nessun attributo
compone	Gruppo(1,N) Solista(0,1)	Un gruppo è composto da uno o n solisti Un solista fa parte di uno o nessun gruppo	Nessun attributo
concesso	Diritto economico(1,N) Distributore(0,N)	Un diritto economico è concesso ad uno o più distributori Un distributore può avere zero o n diritti economici	Data inizio Percentuale ricavo Costo mensile
produce	Produttore(0,N) Prodotto fisico(1,1)	Un produttore produce zero o prodotti fisici Un prodotto fisico è prodotto da un produttore	Numero copie prodotte
venduto	Prodotto fisico(0,N) Editore (0,N)	Un prodotto fisico è venduto a zero o n editori Un editore compra zero o n prodotti fisici	Numero copie vendute
registra	Studio(0,N) Traccia(1,1)	Uno studio registra zero o più tracce Una traccia è registrata in un solo studio	Nessun attributo
occupa	Prenotazione(1,1) Studio(0,N)	Una prenotazione occupa un solo studio Uno studio può essere occupato da zero o n prenotazioni	Nessun attributo
effettua	Prenotazione(1,1) Artista(0,N)	Una artista effettua zero o n prenotazioni Una prenotazione è effettuata da un solo artista	Nessun attributo
determina	Ricevuta(0,1) Prenotazione(0,1)	Una ricevuta può determinare zero o una prenotazione Una prenotazione può determinare zero o una ricevuta.	Nessun attributo
definisce	Ricevuta(1,1) Contratto(0,N)	Ogni ricevuta deve essere definita da un contratto.	Nessun attributo

		Un contratto può non definire nessuna ricevuta.	
stipula	Contratto(1,1) Cliente(1,N)	Un contratto stipulato da un cliente Un cliente stipula da uno ad n contratti	Nessun attributo

### 3.3 Schema ER



## 4. Progettazione logica

### 4.1 Ristrutturazione dello schema

#### 4.1.1 Analisi delle ridondanze

##### 4.1.1.1 Attributo importo nell'entità Ricevuta

L'attributo importo nell'entità Ricevuta può essere ricavato calcolando:

- Il ricavo dalle vendite considerando la quantità di copie vendute moltiplicato per il costo di vendita per singola copia sottratto la quantità di copie prodotte moltiplicato per il costo di produzione di una singola copia.
- Il ricavo dalla concessione dei diritti d'autore sottraendo al costo mensile la percentuale di ricavo decisa dal contratto.

Tavola dei volumi		
Concetto	Tipo	Volume
Ricevuta	Entità	2.000
definisce	Relazione	2.000
Contratto	Entità	400

Stipula (Produttore)	Relazione	20
Produttore	Entità	20
produzione	Relazione	400.000
Prodotto fisico	Entità	400.000
vendita	Relazione	350.000
Distributore	Entità	100
Stipula (Distributore)	Relazione	100
concessione	Relazione	20.000
Diritto d'autore	Entità	10.000

Vengono effettuate le seguenti operazioni in merito all'attributo importo di Ricevuta:

- **Operazione 1** (10 volte al giorno, media mensile): inserimento di una nuova ricevuta relativa alla concessione del diritto d'autore
- **Operazione 2** (4 volte al giorno): visualizzazione di una ricevuta

Le seguenti tabelle riportano il numero di accessi necessari ad eseguire le operazioni sopra riportate, in presenza (attributo presente) e assenza (attributo assente e valore calcolato) dell'attributo importo in Ricevuta.

#### In assenza di ridondanza

Operazione 1			
Concetto	Tipo	Volume	S/L
Ricevuta	Entità	1	S
definisce	Relazione	1	L
Contratto	Entità	1	L
Stipula (Distributore)	Relazione	1	L
Distributore	Entità	1	L
concessione	Relazione	1	L
Diritto d'autore	Entità	1	L

Operazione 2			
Concetto	Tipo	Volume	S/L
Ricevuta	Entità	1	L
definisce	Relazione	1	L
Contratto	Entità	1	L
Stipula (Distributore)	Relazione	1	L
Distributore	Entità	1	L
concessione	Relazione	1	L
Diritto d'autore	Entità	1	L

#### In presenza di ridondanza

Operazione 1			
Concetto	Tipo	Volume	S/L
Ricevuta	Entità	1	S
definisce	Relazione	1	L
Contratto	Entità	1	L
Stipula (Distributore)	Relazione	1	L
Distributore	Entità	1	L
concessione	Relazione	1	L
Diritto d'autore	Entità	1	L

Operazione 2			
Concetto	Tipo	Volume	S/L
Ricevuta	Entità	1	L

Costi in assenza di ridondanza:

- **Operazione 1:** (1 accesso in scrittura + 6 accessi in lettura) x 10 volte al giorno. Totale: 10 accessi in scrittura e 60 accessi in lettura.
- **Operazione 2:** (7 accessi in lettura) x 4 volte al giorno. Totale: 28 accessi in lettura.

Assumendo costo doppio per gli accessi in scrittura, il costo giornaliero è  $(10 \times 2) + 60 + 28 = 108$ .

Costi in presenza di ridondanza:

- **Operazione 1:** (1 accesso in scrittura + 6 accessi in lettura) x 10 volte al giorno. Totale: 10 accessi in scrittura e 60 accessi in lettura.
- **Operazione 2:** (1 accessi in lettura) x 4 volte al giorno. Totale: 4 accessi in lettura.

Assumendo costo doppio per gli accessi in scrittura, il costo giornaliero è  $(10 \times 2) + 60 + 4 = 84$ .

Quindi, considerando le operazioni giornaliere del database e la loro frequenze di esecuzione, è più conveniente tenere l'attributo importo nell'entità Ricevuta piuttosto che calcolare l'importo ogni qualvolta serva ottenerlo.

#### 4.1.1.2 Prodotto, album e singolo

La distinzione di prodotto in album e singolo è basata sul numero di brani dai quali è composto un prodotto: se n. brani > 1 allora è un album. Per riportare la stessa logica presente nello schema ER nello schema ER ristrutturato è possibile sostituire le entità Album e Singolo e la relazione tra esse con una sola relazione ricorsiva. In questo modo però si ha ridondanza poiché la stessa informazione (numero di brani appartenenti ad un determinato prodotto) è già presente nella relazione "forma" tra Brano e Prodotto. È per cui stato deciso di eliminare interamente la suddetta parte e ottenere la differenza tra album e singolo tramite una query.

#### 4.1.1.3 Attributo nome d'arte nelle entità Artista e Solista

L'entità Artista possiede l'attributo nome d'arte per identificare il nome di un artista solista oppure di un gruppo. L'attributo nome d'arte nell'entità Solista invece identifica il nome di ogni persona considerata artista in questo modo è possibile risalire al nome d'arte di ogni componente dei un gruppo musicale. Questo meccanismo implica che nel caso un solista non faccia parte di un gruppo l'attributo nome d'arte in Solista sia uguale all'attributo omonimo in Artista.

#### 4.1.1.4 Attributo titolo nelle entità Brano e Prodotto

L'attributo titolo in Brano identifica il titolo di un singolo brano mentre l'attributo titolo in Prodotto definisce il titolo di un Singolo oppure di un Album. Nel caso in cui il Brano sia un singolo ci sarà una ridondanza perché titolo in Brano e titolo in Prodotto avranno lo stesso valore. Nel caso particolare nel quale il titolo dell'album sia lo stesso di un brano singolo all'interno di esso ci sarà un'altra ridondanza. Queste ridondanze ha senso esistano perché togliendo l'attributo titolo da Brano oppure Prodotto non potremmo risalire al titolo di un Album ma solo ai titoli dei suoi brani.

### 4.1.2 Eliminazione delle generalizzazioni ( ← parziale, ← totale)

Generalizzazione	Risoluzione
Cliente ← Azienda, Artista	È stato deciso di eliminare l'entità cliente poiché contenente solo un attributo. Le figlie hanno per cui ereditato le relazioni del padre.
Artista ← Gruppo, Solista	È stato deciso di eliminare l'entità gruppo e considerare Artista come "gruppo" oppure "solista". Quindi Solista è un'entità che identifica una persona singola che può appartenere ad un gruppo od essere un solo artista.
Azienda ← Produttore, Distributore	È stato deciso di eliminare l'entità azienda poiché Produttore e Distributore sono due concetti diversi con relazioni non simili. Per cui i figli ne ereditano gli attributi e le relazioni.
Distributore ← Editore, Ente telematico	È stato deciso di accorpare le entità figlie a Distributore. Quindi il padre accorperà le relazioni dei figli.
Prodotto ← Album, Singolo	È stato deciso di eliminare le entità e le relazioni figlie. (vedi analisi delle ridondanze 4.1.1.2)
Diritto d'autore ← Diritto Economico, Diritto morale	È stato deciso di accorpare le entità figlie al padre poiché la distinzione tra diritto economico e morale ha senso solo concettualmente. Quindi il Diritto d'autore ha senso come informazione solo quando si ha un prodotto commerciale. Il diritto morale è facilmente ricavabile dall'insieme degli artisti che producono una traccia o un brano.



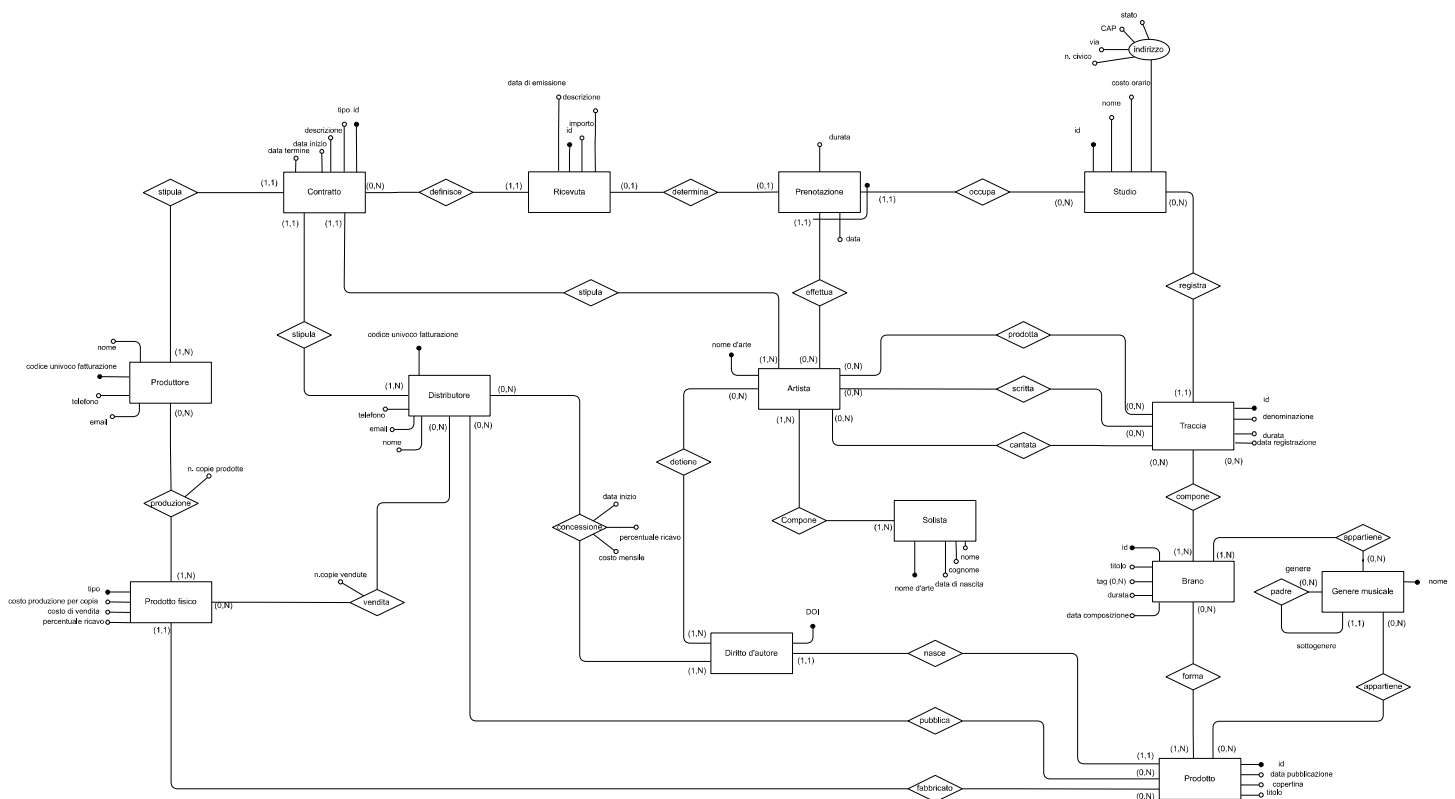
### 4.1.3 Scelta degli identificatori primari

È stato deciso di sostituire alcuni identificatori primari di tipo stringa con degli attributi atomici id di tipo int, questo per ridurre il tempo di elaborazione e ridurre la memoria utilizzata. Le rispettive sostituzioni sono:

- Traccia.denominazione → Traccia.id
- Brano.titolo → Brano.id
- Studio.nome → Studio.id
- Prodotto.titolo → Prodotto.id

Gli attributi chiave primaria usati in precedenza sono diventati dei normali attributi mentre è stata introdotta la nuova chiave primaria di tipo int.

### 4.1.4 Schema ER ristrutturato



## 4.2 Schema relazionale

### 4.2.1 Descrizione schema relazionale e vincoli di integrità referenziale

**Traccia** (id, denominazione, durata, data\_registrazione, studio)

Traccia.studio → Studio.id

**Compone\_bran** (traccia, brano)

Compone\_bran.traccia → Traccia.id

Compone\_bran.bran → Brano.id

**Bran** (id, titolo, durata, data\_composizione)

**Ha\_tag** (brano, nome)

Ha\_tag.bran → Brano.id

Ha\_tag.nome → Tag.nome

**Tag**(nome)

**Appartiene\_bran** (brano, nomeGenere)

Appartiene\_bran.bran → Brano.id

Appartiene\_bran.nomeGenere → Genere\_musicale.nome

**Genere\_musicale** (nome, sottogenere)  
 Genere\_musicale.sottogenere→Genere\_musicale.nome

**Appartiene\_prodotto** (prodotto, nomeGenere)  
 Appartiene\_prodotto.Prodotto→Prodotto.id  
 Appartiene\_prodotto.nomeGenere→Genere\_musicale.nome

**Forma\_prodotto** (brano, prodotto)  
 Forma\_prodotto.Brano→Brano.id  
 Forma\_prodotto.Prodotto→Prodotto.id

**Prodotto** (id, data\_pubblicazione, copertina, titolo, DOI)  
 Prodotto.DOI→Diritto\_autore.DOI

**Diritto\_autore** (DOI)  
**Concessione** (DOI, azienda, data\_inizio, costo\_mensile, percentuale\_ricavo)  
 Concessione.DOI→Diritto\_autore.DOI  
 Concessione.azienda→Distributore.codice\_univoco\_fatturazione

**Pubblica** (azienda, prodotto)  
 Pubblica.azienda→Distributore.codice\_univoco\_fatturazione  
 Pubblica.prodotto→Prodotto.id

**Distributore** (codice\_univoco\_fatturazione, nome, telefono, email)  
**Produttore** (codice\_univoco\_fatturazione, nome, telefono, email)  
**Produzione** (azienda, prodottoFisico, prodotto, n\_copie\_prodotte)  
 Produzione.azienda→Produttore.codice\_univoco\_fatturazione  
 Produzione.prodottoFisico→Prodotto\_fisico.tipo  
 Produzione.prodotto→Prodotto.id

**Prodotto\_fisico** (tipo, prodotto, costo\_produzione\_per\_copia, costo\_vendita, percentuale\_ricavo)  
**Vendita** (azienda, prodottoFisico, prodotto, n\_copie\_vendute)  
 Vendita.azienda→Distributore.codice\_univoco\_fatturazione  
 Vendita.prodottoFisico→Prodotto\_fisico.tipo  
 Vendita.prodotto→Prodotto.id

**Detiene** (DOI, artista)  
 Detiene.DOI→Diritto\_autore.DOI  
 Detiene.artista→Artista.nome\_arte

**Artista** (nome\_arte)  
**Compone\_artista** (artista, solista)  
 Compone\_artista.artista→Artista.nome\_arte  
 Compone\_artista.solista→Solista.nome\_arte

**Solista** (nome\_arte, nome, cognome, data\_nascita)  
**Prodotta** (artista, traccia)  
 Prodotta.artista→Artista.nome\_arte  
 Prodotta.traccia→Traccia.id

**Scritta** (artista, traccia)  
 Scritta.artista→Artista.nome\_arte  
 Scritta.traccia→Traccia.id

**Cantata** (artista, traccia)  
 Cantata.artista→Artista.nome\_arte  
 Cantata.traccia→Traccia.id

**Studio** (id, nome, costo\_orario, via, n\_civico, CAP, stato)  
**Prenotazione** (data, studio, nome\_artista, durata, ricevuta)  
 Prenotazione.studio→Studio.id  
 Prenotazione.nome\_artista→Artista.nome\_arte  
 Prenotazione.ricevuta→Ricevuta.id

**Contratto** (id, tipo, descrizione, data\_inizio, data\_termine, produttore, distributore, artista)  
 Contratto.produttore→Produttore.codice\_univoco\_fatturazione  
 Contratto.distributore→Distributore.codice\_univoco\_fatturazione  
 Contratto.artista→Artista.nome\_arte

**Ricevuta** (id, importo, descrizione, data\_emissione, contratto)  
 Ricevuta.contratto→Contratto.id

## 5. Query e indici

### 5.1 Queries

**Query 1:** Visualizza l'importo totale guadagnato da un prodotto. (costo di vendita - costo di produzione - percentuale per artista).

```
1 select titolo as prodotto, sum(vendita_totale - (vendita_totale * percentuale_ricavo) - costo_totale) as totale
2 from (
3     select titolo, percentuale_ricavo,
4           sum(costo_vendita*n_copie_vendute) as vendita_totale,
5           sum(costo_produzione_per_copia*n_copie_prodotte) as costo_totale
6     from prodotto
7     join prodotto_fisico on (prodotto_fisico.prodotto = prodotto.id)
8     join vendita on (prodotto_fisico.prodotto = vendita.prodotto
9                     and prodotto_fisico.tipo = vendita.prodottofisico)
10    join produzione on (prodotto_fisico.prodotto = produzione.prodotto
11                      and prodotto_fisico.tipo = produzione.prodottofisico)
12   group by titolo, percentuale_ricavo
13 ) as costi
14 group by prodotto
15 order by totale desc
```

	prodotto character varying (50)	totale money
1	TrapKing	14.796,00 €
2	Sponge	12.040,00 €

**Query 2:** Visualizza, per ogni genere, l'artista che ha più brani appartenenti a quel genere musicale.

```
1 drop view if exists artisti_per_canzone;
2 create view artisti_per_canzone as
3 select nome_arte as artista, brano.id as brano, titolo
4 from brano
5     join compone_branco on (brano.id = compone_branco.branco)
6     join traccia on (traccia.id = compone_branco.traccia)
7     left join prodotta on (traccia.id = prodotta.traccia)
8     join artista on (artista.nome_arte = prodotta.artista)
9     left join scritta on (artista.nome_arte = scritta.artista)
10    left join cantata on (artista.nome_arte = cantata.artista)
11 group by nome_arte, brano.id, titolo;
12
13 select distinct nome, artista, max(totale) as massimo
14 from (
15     select artista, nome, count(titolo) as totale
16     from artisti_per_canzone
17     join appartiene_branco on (artisti_per_canzone.branco = appartiene_branco.branco)
18     join genere_musicale on (genere_musicale.nome = appartiene_branco.nomegenere)
19     group by artista, nome
20 ) as generi_canzoni
21 group by nome, artista
```

	nome character varying (30)	artista character varying (30)	massimo bigint
1	drill	Dark Polo	1
2	drill	Giorgi	1
3	elettronica	Travis Scotty	1
4	punk rock	Daft Funk	2
5	techno	Dark Polo	1
6	trap	Dark Polo	1
7	trap	Giorgi	1
8	trap	Travis Scotty	1

**Query 3:** Visualizza il guadagno totale di un'artista. (guadagno concessioni + guadagno vendite - costo prenotazioni)

```

1 select ricavi_concessione.artista, ricavo_concessioni + coalesce(ricavo_vendita, 0::money) - coalesce(costi_prenotazioni, 0::money) as guadagno
2 from (
3     select nome_arte as artista, sum(((DATE_PART('year', CURRENT_DATE) - DATE_PART('year', concessione.data_inizio)) * 12 +
4         (DATE_PART('month', CURRENT_DATE) - DATE_PART('month', concessione.data_inizio))) * (costo_mensile * percentuale_ricavo)) as ricavo_concessioni
5     from artista
6     join detiene on (artista.nome_arte = detiene.artista)
7     join diritto_autore on (detiene.doi = diritto_autore.doi)
8     join concessione on (diritto_autore.doi = concessione.doi)
9     group by nome_arte
10 ) as ricavi_concessione
11 left join (
12     select nome_arte as artista, sum(ricevuta.importo) as costi_prenotazioni
13     from artista
14     join prenotazione on (prenotazione.artista = artista.nome_arte)
15     join ricevuta on (ricevuta.id = prenotazione.ricevuta)
16     group by nome_arte
17 ) as costi_prenotazioni on (ricavi_concessione.artista = costi_prenotazioni.artista)
18 left join (
19     select artista, sum(costo_vendita*n_copie_vendute*percentuale_ricavo) as ricavo_vendita
20     from prodotto
21     join prodotto_fisico on (prodotto_fisico.prodotto = prodotto.id)
22     join vendita on (prodotto_fisico.prodotto = vendita.prodotto
23         and prodotto_fisico.tipo = vendita.prodottofisico)
24     join(
25         select nome_arte as artista, forma_prodotto.prodotto as prodotto
26         from forma_prodotto
27         join compone_branco on (compone_branco.branco = forma_prodotto.branco)
28         left join prodotta on (compone_branco.traccia = prodotta.traccia)
29         join artista on (artista.nome_arte = prodotta.artista)
30         left join scritta on (artista.nome_arte = scritta.artista)
31         left join cantata on (artista.nome_arte = cantata.artista)
32         group by nome_arte, forma_prodotto.prodotto
33     ) as art_prod on (prodotto.id = art_prod.prodotto)
34     group by artista
35 ) as ricavi_vendita on (ricavi_vendita.artista = ricavi_concessione.artista)
36 order by guadagno desc

```

	artista character varying (30)	guadagno money
1	TonyF	46.620,00 €
2	Travis Scotty	37.229,00 €
3	Dark Polo	31.342,00 €
4	Giorgi	12.944,00 €
5	Daft Funk	12.934,00 €
6	Claudio Bisios	8.100,00 €

**Query 4:** Visualizza l'artista che ha creato più prodotti.

```

1 drop view if exists artisti_per_prodotto;
2 create view artisti_per_prodotto as
3 select nome_arte as artista, compone_branco.branco as brano
4 from forma_prodotto
5     join compone_branco on (compone_branco.branco = forma_prodotto.branco)
6     left join prodotta on (compone_branco.traccia = prodotta.traccia)
7     join artista on (artista.nome_arte = prodotta.artista)
8     left join scritta on (artista.nome_arte = scritta.artista)
9     left join cantata on (artista.nome_arte = cantata.artista)
10 group by nome_arte, compone_branco.branco;
11
12 select massimo_prodotti.artista, massimo_prodotti.massimo
13 from(
14     select artista, max(numero_prodotti) as massimo
15     from(
16         select artista, count(branco) as numero_prodotti
17         from artisti_per_prodotto
18         group by artista
19     ) as conto_prodotti
20     group by artista
21 ) as massimo_prodotti
22 where massimo_prodotti.massimo = (
23     select max(numero_prodotti)
24     from(
25         select artista, count(branco) as numero_prodotti
26         from artisti_per_prodotto
27         group by artista
28     ) as conto_prodotti
29 )
30 group by artista, massimo_prodotti.massimo
31 order by artista asc

```

	artista character varying (30)	massimo bigint
1	Daft Funk	2
2	Dark Polo	2
3	Travis Scotty	2

**Query 5:** Visualizzare i prodotti che hanno un maggior rapporto copie vendute/copie prodotte.

```

1 select prodotto.titolo as prodotto, to_char((cast((sum(n_copie_vendute))as float)
2                                     /cast((sum(n_copie_prodotte))as float)), '0.999') as rapporto
3 from prodotto
4     join prodotto_fisico on (prodotto.id = prodotto_fisico.prodotto)
5     join vendita on (prodotto_fisico.prodotto = vendita.prodotto
6                     and prodotto_fisico.tipo = vendita.prodottofisico)
7     join produzione on (prodotto_fisico.prodotto = produzione.prodotto
8                       and prodotto_fisico.tipo = produzione.prodottofisico)
9 group by prodotto.titolo
10 order by rapporto desc

```

	prodotto character varying (50)	rapporto text
1	TrapKing	0.760
2	Sponge	0.629

**Query 6:** Visualizzare il distributore che ha acquistato il maggior numero di prodotti fisici per ogni prodotto.

```

1 select v1.prodotto, v1.azienda, v1.massimo as copie_acquistate
2 from (
3     select totale_copie.prodotto as prodotto, totale_copie.azienda, max(totale_copie.numero_copie) as massimo
4     from(
5         select prodotto.titolo as prodotto, distributore.nome as azienda, sum(n_copie_vendute) as numero_copie
6         from prodotto
7             join prodotto_fisico on (prodotto.id = prodotto_fisico.prodotto)
8             join vendita on (prodotto_fisico.prodotto = vendita.prodotto
9                             and prodotto_fisico.tipo = vendita.prodottofisico)
10            join distributore on (vendita.azienda = distributore.codice_univoco_fatturazione)
11        group by prodotto.titolo, distributore.nome
12    ) as totale_copie
13    group by totale_copie.prodotto, totale_copie.azienda
14 ) as v1
15 join (
16     select totale_copie.prodotto as prodotto, max(totale_copie.numero_copie) as massimo
17     from(
18         select prodotto.titolo as prodotto, distributore.nome as azienda, sum(n_copie_vendute) as numero_copie
19         from prodotto
20             join prodotto_fisico on (prodotto.id = prodotto_fisico.prodotto)
21             join vendita on (prodotto_fisico.prodotto = vendita.prodotto
22                             and prodotto_fisico.tipo = vendita.prodottofisico)
23            join distributore on (vendita.azienda = distributore.codice_univoco_fatturazione)
24        group by prodotto.titolo, distributore.nome
25    ) as totale_copie
26    group by totale_copie.prodotto
27 ) as v2
28 on (v1.massimo = v2.massimo)
29 order by copie_acquistate desc

```

	prodotto character varying (50)	azienda character varying (100)	copie_acquistate bigint
1	Sponge	Filtrinelli	19000
2	TrapKing	Distribu	17000

## 5.2 Indici

### 5.2.1 Indice per la ricerca delle tracce

Una funzione importante per una casa discografica è la possibilità di effettuare una ricerca rapida delle tracce che sono state registrate nei suoi studi. È infatti fondamentale, nel processo di editing e montaggio di un brano, riuscire a trovare le tracce inerenti in poco tempo.

Siccome il nome della traccia cambia raramente e contiene, a discrezione del creatore della traccia, delle parole chiave è stato deciso di creare un indice per la ricerca che considera solo il campo denominazione della tabella Traccia.

```

1 drop index if exists idx_tracce;
2 create index idx_tracce on Traccia (denominazione);

```

**Query di esempio:** visualizzare tutte le tracce che sono basi musicali (con parola chiave “base”). Si vuole simulare la ricerca che un producer può effettuare durante il processo di produzione, prendendo come esempio che le tracce inerenti al brano e contenenti la base o le basi musicali siano state salvate durante le registrazioni con la parola chiave “base”.

```

1 select * from traccia where lower(denominazione) like '%base%';

```

Prima della creazione dell'indice la query di esempio impiegava 129 millisecondi, dopo la creazione dell'indice la query impiega 72 millisecondi. Un miglioramento del 55%.



## 6. Codice c++

### 6.1 Interfaccia grafica

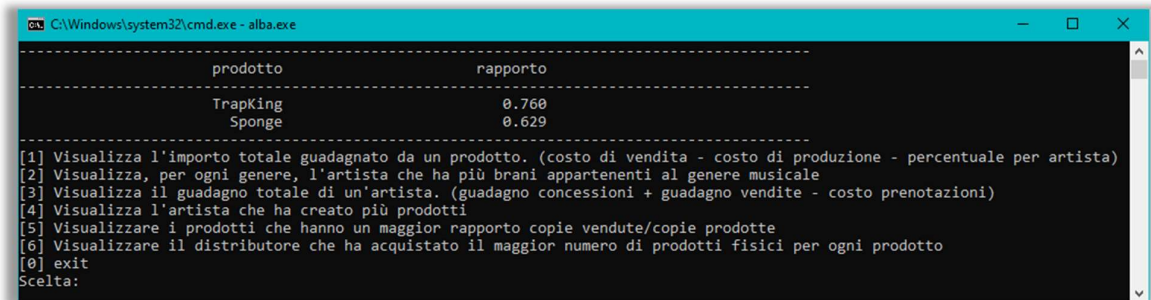
Il programma c++ permette di eseguire le query tramite un'interfaccia grafica a terminale.

L'interfaccia è spartana ma chiara. Inizialmente viene stampato il menu di selezione che presenta le query che possono essere eseguite con il loro corrispondente numero di selezione.



```
C:\Windows\system32\cmd.exe - alba.exe
C:\Users\luca\Google Drive\Scuola\Uni\SC1167\BASI DI DATI\Progetto\Progetto basi di dati\codice>alba.exe
Connessione avvenuta correttamente
[1] Visualizza l'importo totale guadagnato da un prodotto. (costo di vendita - costo di produzione - percentuale per artista)
[2] Visualizza, per ogni genere, l'artista che ha più brani appartenenti al genere musicale
[3] Visualizza il guadagno totale di un'artista. (guadagno concessioni + guadagno vendite - costo prenotazioni)
[4] Visualizza l'artista che ha creato più prodotti
[5] Visualizzare i prodotti che hanno un maggior rapporto copie vendute/copie prodotte
[6] Visualizzare il distributore che ha acquistato il maggior numero di prodotti fisici per ogni prodotto
[0] exit
Scelta:
```

Selezionando, ad esempio, la query numero 5 il programma stamperà a video i risultati e nuovamente il menu di selezione. Ogni volta che si esegue questa operazione il programma elimina i risultati presentati a video precedentemente.



```
C:\Windows\system32\cmd.exe - alba.exe
-----
          prodotto          rapporto
-----
          TrapKing          0.760
          Sponge            0.629
-----
[1] Visualizza l'importo totale guadagnato da un prodotto. (costo di vendita - costo di produzione - percentuale per artista)
[2] Visualizza, per ogni genere, l'artista che ha più brani appartenenti al genere musicale
[3] Visualizza il guadagno totale di un'artista. (guadagno concessioni + guadagno vendite - costo prenotazioni)
[4] Visualizza l'artista che ha creato più prodotti
[5] Visualizzare i prodotti che hanno un maggior rapporto copie vendute/copie prodotte
[6] Visualizzare il distributore che ha acquistato il maggior numero di prodotti fisici per ogni prodotto
[0] exit
Scelta:
```

### 6.2 Codice

Il codice è contenuto in un unico file .cpp. Ogni query e le relative operazioni sono contenute in funzioni separate, tutte con la stessa struttura: esecuzione della query contenuta in una stringa, salvataggio e stampa del risultato.

La funzione main usa un ciclo do while per la presentazione del menu di selezione, l'input della scelta e un costrutto switch per la selezione della query. Il ciclo principale termina solo quando l'input è uguale a 0.

Per compilare il programma è stato usato il compilatore MinGW su sistema operativo windows e la libreria **lpq**. I file usati, nelle rispettive cartelle, sono quindi:

- dependencies/lib: **libpq.lib**, **libpq.dll**
- dependencies/include: **libpq-fe.h**, **pq\_config\_ext.h**, **postgres\_ext.h**

Il comando utilizzato per effettuare la compilazione è:

```
g++ alba.cpp -L dependencies/lib -lpq -o alba
```